

# Auto Scaling Group with Load Balancer

## Introduction

In this project, I created a scalable and reliable web infrastructure on AWS. I used an Application Load Balancer to evenly distribute incoming website traffic and Auto Scaling Groups to automatically manage the number of active servers.

The three Auto Scaling Groups are set up with different scaling strategies:

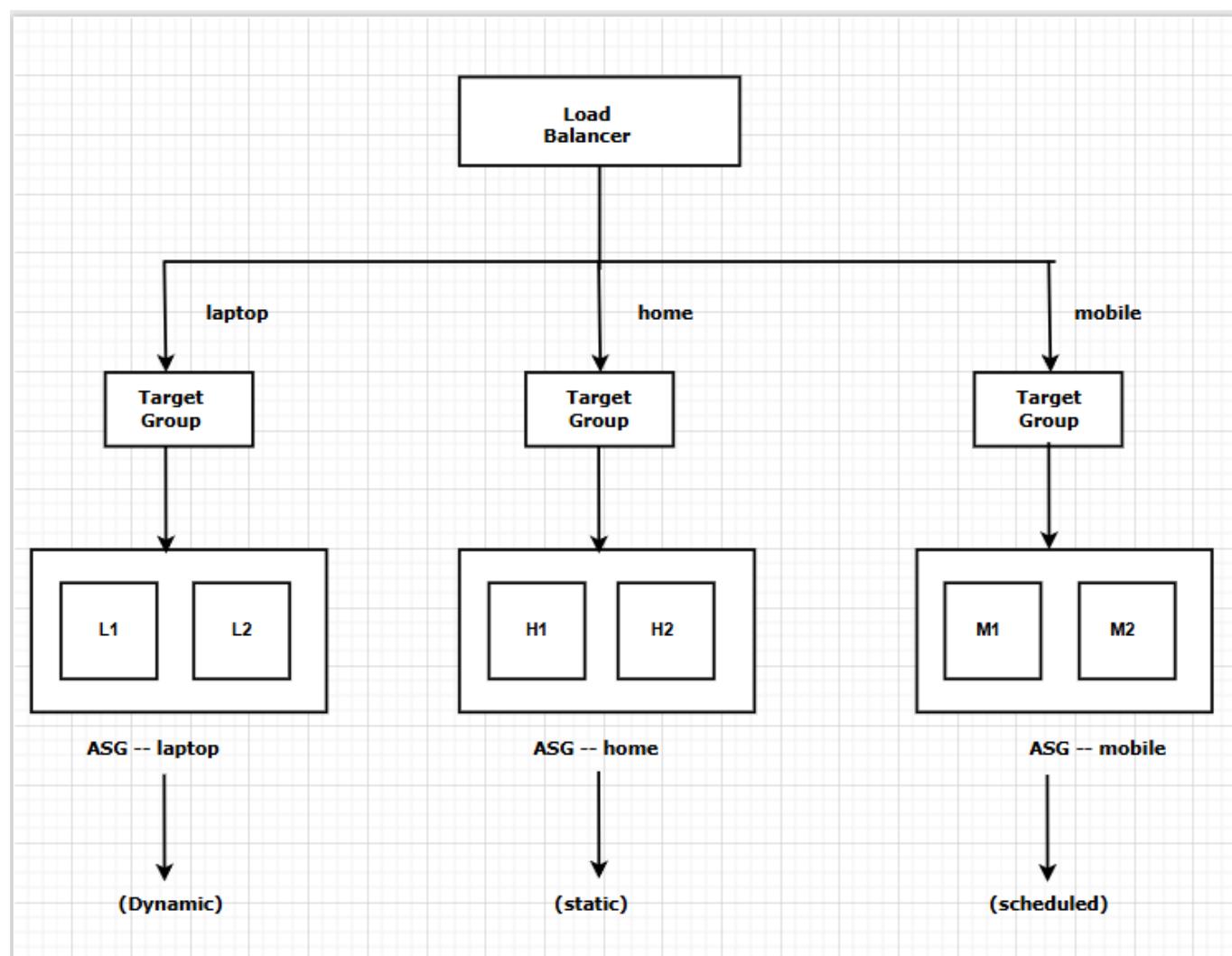
Home: Maintains a constant, static number of servers.

Mobile: Scales based on a schedule (e.g., adding servers during peak business hours).

Laptop: Scales dynamically based on real-time CPU usage.

Each group has its own target group, which is registered with the central Load Balancer to handle the routing.

## Architecture Diagram



## Features Implemented

## -- Application Load Balancer

Acts like a traffic manager for your website.

Sends incoming visitors to different groups of servers to spread out the load.

Makes sure the website stays online even if one server has a problem.

-- Auto Scaling Groups I set up three groups that automatically control how many servers are running:

Home (Static): Always keeps the same number of servers on.

Mobile (Scheduled): Adds more servers at specific busy times (like lunch hour) and removes them later.

Laptop (Dynamic): Automatically adds servers when traffic is high and removes them when it's quiet.

## -- Target Groups

Each of the three server groups above has its own "Target Group."

The Load Balancer knows about all these groups and can send visitors to the right one.

# Deployment Steps

## Step1 : Launch Template

Launch 3 Templates : Home , Mobile and laptop with diffrent user data script.

Go to Launch Template click on create template

Name your template

Write Description

Select AMI from Quick Start (I choosed Amazon Linux) You can choose your own AMI if u have created one.

**Application and OS Images (Amazon Machine Image)** [Info](#)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose [Browse more AMIs](#).

[Search our full catalog including 1000s of application and OS images](#)

**Recents** **Quick Start**

Don't include in launch template | Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | D | [View all AMIs](#)

**Amazon Machine Image (AMI)**

Amazon Linux 2023 kernel-6.1 AMI | [Free tier eligible](#)

ami-08982f1c5bf93d976 (64-bit (x86), uefi-preferred) / ami-039f81f5ce6752b10 (64-bit (Arm), uefi)  
Virtualization: hvm ENA enabled: true Root device type: ebs

**Description**

Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.8.20250915.0 x86\_64 HVM kernel-6.1

**Summary**

**Software Image (AMI)**  
Amazon Linux 2023 AMI 2023.8.2...[read more](#)  
ami-08982f1c5bf93d976

**Virtual server type (instance type)**  
t3.micro

**Firewall (security group)**  
launch-wizard-1

**Storage (volumes)**  
1 volume(s) - 8 GiB

[Cancel](#) [Create launch template](#)

## Select Instance Type

## Select Keypair

**Instance type** [Info](#) | [Get advice](#)

**Instance type**

t3.micro | [Free tier eligible](#)

Family: t3 2 vCPU 1 GiB Memory Current generation: true  
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour  
On-Demand SUSE base pricing: 0.0104 USD per Hour  
On-Demand Linux base pricing: 0.0104 USD per Hour  
On-Demand RHEL base pricing: 0.0392 USD per Hour  
On-Demand Windows base pricing: 0.0196 USD per Hour

**Additional costs apply for AMIs with pre-installed software**

**Advanced**

All generations | [Compare instance types](#)

**Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

**Key pair name**

second-key | [Create new key pair](#)

**Network settings** [Info](#)

**Subnet** | [Info](#)

Don't include in launch template | [Create new subnet](#)

**Summary**

**Software Image (AMI)**  
Amazon Linux 2023 AMI 2023.8.2...[read more](#)  
ami-08982f1c5bf93d976

**Virtual server type (instance type)**  
t3.micro

**Firewall (security group)**  
launch-wizard-1

**Storage (volumes)**  
1 volume(s) - 8 GiB

[Cancel](#) [Create launch template](#)

Select Network Settings. here my application is simple so i only need port 80 and port 22. you can select existing security group too.

add user datascript in advanced options. (you don't need to add this script if u haе selected your own AMI )

click on create template

Similarly Create other templates for Mobile and Laptop

User datascript for mobile :

```
#!/bin/bash
sudo yum update -y
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd

sudo mkdir /var/www/html/mobile
echo "<h1>This is the mobile page $(hostname -f)</h1>" >
/var/www/html/mobile/index.html
```

User datascript for laptop :

```
#!/bin/bash
sudo yum update -y
sudo yum install -y httpd
sudo systemctl start httpd
sudo systemctl enable httpd

sudo mkdir /var/www/html/laptop
echo "<h1>This is the laptop page $(hostname -f)</h1>" >
/var/www/html/laptop/index.html
```

## Step2 : Creating AutoScalling Group (ASG)

Let's Start with creating ASG for mobile. It is a Sceduled ASG.

Go to Autoscalling Group and click on create autoscalling.

(You will see the 7 steps on lefthand side of the screen)

### 1. Choose Launch Template

Give Name to autoscalling group.

Select a Launch Template of mobile for mobile autoscalling group.

The screenshot shows the AWS EC2 Auto Scaling Groups 'Create Auto Scaling group' wizard. The current step is 'Step 1: Choose launch template'. On the left, there is a vertical list of steps: Step 1 (radio button selected), Step 2, Step 3 - optional, Step 4 - optional, Step 5 - optional, Step 6 - optional, Step 7, and Review. The main area has two sections: 'Choose launch template' and 'Launch template'. In the 'Choose launch template' section, it says 'Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.' Below this is a 'Name' field with 'Auto Scaling group name' and a text input box containing 'Home-ASG'. A note below says 'Must be unique to this account in the current Region and no more than 255 characters.' In the 'Launch template' section, it says 'Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.' Below this is a dropdown menu showing 'Home-LT' selected, with a note: 'For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.' At the bottom, there are buttons for 'Create' and 'Next Step'.

click on next...go to next step..



### 2. Choose instance launch option

Select Availability Zone. (Atleast 2)

go to next step..

VPC  
Choose the VPC that defines the virtual network for your Auto Scaling group.  
vp-06e769ac5be1c7149  
172.31.0.0/16 Default

Create a VPC [\[?\]](#)

Availability Zones and subnets  
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets [\[?\]](#)

use1-az1 (us-east-1a) | subnet-0e01aebcd480799  
172.31.0.0/20 Default

use1-az4 (us-east-1c) | subnet-03b9c870019d28e78  
172.31.16.0/20 Default

Create a subnet [\[?\]](#)

Availability Zone distribution - new  
Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

Balanced best effort  
If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.

Balanced only  
If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

Cancel Skip to review Previous Next

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

### 3. Integrate with other services

We are going to integrate these ASG with Loadbalancer after creating all 3 ASG. So for now let's go to next step..

Step 1 Choose launch template

Step 2 Choose instance launch options

Step 3 - optional **Integrate with other services**

Step 4 - optional Configure group size and scaling

Step 5 - optional Add notifications

Step 6 - optional Add tags

Step 7 Review

**Integrate with other services - optional** [Info](#)  
Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.

**Load balancing** [Info](#)  
Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer  
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer  
Choose from your existing load balancers.

Attach to a new load balancer  
Quickly create a basic load balancer to attach to your Auto Scaling group.

**VPC Lattice integration options** [Info](#)  
To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

**Select VPC Lattice service to attach**

No VPC Lattice service  
VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

Attach to VPC Lattice service  
Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

Create new VPC Lattice service [\[?\]](#)

**Application Recovery Controller (ARC) zonal shift - new** [Info](#)  
During an Availability Zone impairment, target instance launches towards other healthy Availability Zones.

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

### 4. Configure Group size and scalling

Here for mobile we are giving : Desired = 3 , min = 2 , max = 7. (After creating mobile ASG we will make it sceduled)

For Laptop (Dynamic) ASG select : Desired = 3 , min = 2 , max = 7.

For Home (static) ASG select : Desired = 2 , min = 2 , max = 2.

The screenshot shows the AWS EC2 Auto Scaling group creation wizard at Step 4: Configure group size and scaling. The left sidebar lists steps from 1 to 7, with step 4 highlighted. The main content area is titled "Configure group size and scaling - optional". It includes sections for "Group size" (info) and "Scaling" (info). Under "Group size", the "Desired capacity type" is set to "Units (number of instances)" with a value of 2. Under "Scaling", "Min desired capacity" and "Max desired capacity" are both set to 2, with the note "Equal or less than desired capacity". There is also a section for "Automatic scaling - optional" with a link to "Choose whether to use a target tracking policy".

select target tracking scaling policy ( if your ASG is static you don't need policy so select No scaling policy )

select metric type - Here i choose CPU utilization.

select target value - I choose 50.

select instance warmup - i choose 300 sec i.e, 5 min.

go to next step..

## 5. Add notification

The screenshot shows the AWS EC2 Auto Scaling group creation wizard at Step 5: Add notifications. The left sidebar lists steps from 1 to 7, with step 5 highlighted. The main content area is titled "Add notifications - optional". It includes a "Add notification" button. At the bottom right, there are buttons for "Cancel", "Skip to review", "Previous", and "Next".

## 6. Add tags

Step 1  
Choose launch template  
Step 2  
Choose instance launch options  
Step 3 - optional  
Configure group size and scaling  
Step 4 - optional  
Step 5 - optional  
Step 6 - optional  
**Add tags**  
Step 7  
Review

**Add tags - optional** Info

Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

You can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group.

**Tags (0)**

Add tag

50 remaining

Cancel Previous Next

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

you don't have to do anything in step 5 and 6 so just click on next.

## 7. Review

select create autoscaling group

Instance scale-in protection Disabled	Monitoring Disabled	Default instance warmup Disabled
--	------------------------	-------------------------------------

**Capacity Reservation preference**

Preference Default Capacity Reservation IDs - Resource Groups -

**Step 5: Add notifications**

**Notifications**  
No notifications

**Step 6: Add tags**

**Tags (0)**

Key Value Tag new instances

No tags

Preview code Cancel Previous Create Auto Scaling group

Similarly create other 2 ASG according to their type static and dynamic. Just need to change the group size as i mentioned.

**Auto Scaling groups (3) Info**

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones	Creation time
Mobile-ASG	Mobile-LT   Version Default	0	Updating capacity...	3	2	7	2 Availability Zones	Sat Sep 27 2025 1...
Laptop-ASG	Laptop-LT   Version Default	3	-	3	2	7	2 Availability Zones	Sat Sep 27 2025 1...
Home-ASG	Home-LT   Version Default	2	-	2	2	2	2 Availability Zones	Sat Sep 27 2025 1...

0 Auto Scaling groups selected

## 8. Now to make Mobile ASG Scheduled follow following steps :

Click on Mobile ASG...(You will see the following interface)

**Auto Scaling group: Mobile-ASG**

Automatic scaling

Scaling policies resize your Auto Scaling group to meet changes in demand. With reactive dynamic scaling policies, you can track specific CloudWatch metrics and take action when the CloudWatch alarm threshold is met. Use predictive scaling policies along with dynamic scaling policies in the following situations: when your application demand changes quickly, but with a recurring pattern, or when your EC2 instances require more time to initialize.

**Dynamic scaling policies (1) Info**

Actions Create dynamic scaling policy

Scroll down to bottom..

Click on Create schedule action..

**Auto Scaling groups (1/3) Info**

Name	Last updated	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability
Mobile-ASG	1 minute ago	Mobile-LT   Version Default	3	-	3	2	7	2 Available
Laptop-ASG		Laptop-LT   Version Default	3	-	3	2	7	2 Available

**Scheduled actions (0) Info**

Name	Start time	End time	Recurrence	Time zone	Desired capac...	Min	Max
------	------------	----------	------------	-----------	------------------	-----	-----

No scheduled actions are currently specified

[Create scheduled action](#)

Give name to your schedule.

Enter Desired , min and max values of instances.

Choose Cron option in recurrence and give specific time in side box ( sequence is - minute hour date month and day of week)

Provide at least one value for Desired, Min, or Max Capacity

Desired capacity: 8    Min: 5    Max: 15

Recurrence: Cron    0 10 21 10 \*

Time zone: Etc/UTC

Specific start time: 00:00    YYYY/MM/DD:    Etc/UTC

End by: 2025/10/30    10:30    Etc/UTC

Enter the ending time with date

click on Create

If you click on mobile ASG and scroll down to Scheduled Action you will see the the Created scheduled action.

Step3 : Creating Target Group For each Autoscalling Group

Go to Target Groups Click on Create Target Group

The screenshot shows the AWS EC2 Target groups page. The left sidebar includes sections for Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), and Load Balancing (Load Balancers, Target Groups, Trust Stores). The main content area displays a table header for 'Target groups' with columns for Name, ARN, Port, Protocol, Target type, Load balancer, and VPC ID. A message indicates 'No target groups' and provides a 'Create target group' button.

Give a Name to your Target Group.

The screenshot shows the 'Create target group' wizard. Step 1: Set target group name. It shows a field with 'Home-TG' entered. Other fields include 'Protocol' (HTTP selected), 'Port' (80), 'IP address type' (IPv4 selected), 'VPC' (vpc-06e769ac5be1c7149 selected), and 'Protocol version' (HTTP1 selected).

Scroll down and go to health check section.

Give a path of your application in Health Check Path section.

[Note : i Give "/" as a path . Here "/" represent the directory "/var/www/html/" which is default directory of httpd]

Go to next step.

**Health checks**  
The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

**Health check protocol**  
HTTP

**Health check path**  
Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.  
/

**Advanced health check settings**

**Attributes**  
Certain default attributes will be applied to your target group. You can view and edit them after creating the target group.

**Tags - optional**  
Consider adding tags to your target group. Tags enable you to categorize your AWS resources so you can more easily manage them.

Cancel Next

Click on Create Target Group (don't need to change anything else)

**Ports for the selected instances**  
Ports for routing traffic to the selected instances.  
80  
1-65535 (separate multiple ports with commas)

**Review targets**

**Targets (0)**  
No instances added yet  
Specify instances above, or leave the group empty if you prefer to add targets later.

0 pending Cancel Previous Create target group

--Similarly Create Target Groups for Laptop and Mobile.

You only need to change the path in Health check section as given.

For Laptop :

The screenshot shows the 'Create target group' wizard on the AWS EC2 service. The first step, 'Basic settings', is displayed. Key configuration includes:

- Target group name:** Laptop-TG
- Protocol:** HTTP
- Port:** 80
- IP address type:** IPv4 (selected)
- VPC:** vpc-06e769ac5be1c7149 (selected)
- Protocol version:** HTTP1

The screenshot shows the 'Create target group' wizard on the AWS EC2 service. The second step, 'Advanced settings', is displayed. Key configuration includes:

- Health checks:** Enabled.
- Health check protocol:** HTTP
- Health check path:** /laptop/
- Advanced health check settings:** Available via a link.
- Attributes:** A note states certain default attributes will be applied.
- Tags - optional:** A note encourages adding tags for resource management.

For Mobile :

The screenshot shows the 'Create target group' wizard on the AWS EC2 service. The first step is to define the target group details:

- Target group name:** Mobile\_TG
- Protocol:** HTTP
- Port:** 80
- IP address type:** IPv4
- VPC:** A specific VPC is selected.
- Protocol version:** HTTP1

At the bottom, there are links for CloudShell, Feedback, and standard footer links.

The screenshot shows the continuation of the 'Create target group' wizard, moving to the 'Health checks' section:

- Health check protocol:** HTTP
- Health check path:** /mobile/
- Advanced health check settings:** An expandable section.
- Attributes:** A note about default attributes.
- Tags - optional:** A note about adding tags.

At the bottom right are 'Cancel' and 'Next' buttons.

The 3 Target Groups are created.

**Target groups (3) Info**

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
Mobile-TG	arn:aws:elasticloadbalancing:us-east-1:658394313504:targetgroup/Mobile-TG/2d669565-ac90-42fe-97c1-849a4ee7cd2c	80	HTTP	Instance	None associated	vpc-06e769ac5be1c7149
Laptop-TG	arn:aws:elasticloadbalancing:us-east-1:658394313504:targetgroup/Laptop-TG/2d669565-ac90-42fe-97c1-849a4ee7cd2c	80	HTTP	Instance	None associated	vpc-06e769ac5be1c7149
Home-TG	arn:aws:elasticloadbalancing:us-east-1:658394313504:targetgroup/Home-TG/2d669565-ac90-42fe-97c1-849a4ee7cd2c	80	HTTP	Instance	None associated	vpc-06e769ac5be1c7149

**0 target groups selected**

Select a target group above.

## Step4 : Connecting Autoscalling group to Target Goups

Select an Autoscalling Group .

Click on Action at top right corner.

Click on Edit

**Auto Scaling groups (1/3) Info**

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability
Mobile-ASG	Mobile-LT   Version Default	3	-	3	2	7	2 Availability
Laptop-ASG	Laptop-LT   Version Default	3	-	3	2	7	2 Availability
Home-ASG	Home-LT   Version Default	2	-	2	2	2	2 Availability

**Auto Scaling group: Mobile-ASG**

**Details** Integrations - new Automatic scaling Instance management Instance refresh Activity Monitoring

**Mobile-ASG Capacity overview**

Desired capacity 3	Scaling limits (Min - Max) 2 - 7	Desired capacity type Units (number of instances)	Status -
-----------------------	-------------------------------------	--	-------------

Date created  
Sat Sep 27 2025 18:46:53 GMT+0530 (India Standard Time)

Scroll down and you will see the Load balancing section.

Select a Target group for autoscalling group we selected. (for laptop ASG select Lptop target group, for mobile ASG select Mobile Target group.)

**Load balancing - optional**

**Load balancers**

Application, Network or Gateway Load Balancer target groups  
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

Mobile-TG | HTTP X

One of your target groups is not yet associated with any load balancer. In order for routing and scaling to occur, you will need to attach the target group to a load balancer. This can be done later in the [Load Balancing console](#).

Classic Load Balancers

Create and attach new load balancers

Add a new load balancer

**VPC Lattice integration options - optional**

Enter Cooldown time and click on update.

Successfully connected the Autoscalling group and target groups.

Repeat process to connect each autoscalling group to its target group.

1 Default

Add policy

Suspended processes [Info](#)  
Select suspended processes

Maximum instance lifetime  
 seconds

Default cooldown [Info](#)  
300 seconds

Default instance warmup [Info](#)  
The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.  
 Enable default instance warmup

Tags (0)

Add tag

50 remaining

Cancel Update

Step5 : Creating Application Load Balancer and Connecting to Target Groups.

Go to Load Balancers and Click on Create Load Balancer

The screenshot shows the AWS EC2 Load Balancers page. On the left, there's a sidebar with navigation links for Images, Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, and Settings. The main area has a heading "Load balancers" with a sub-instruction: "Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic." Below this is a search bar labeled "Filter load balancers" and a table header with columns: Name, State, Type, Scheme, IP address type, VPC ID, Availability Zones, and Security. A cartoon robot is sitting at a desk with a computer monitor displaying a cloud icon. A message at the bottom says "0 load balancers selected" and "Select a load balancer above." At the bottom right, there are links for CloudShell, Feedback, and copyright information.

Select Application Load Balancer. Click on Create

The screenshot shows the "Compare and select load balancer type" page. It features three cards: "Application Load Balancer" (Info), "Network Load Balancer" (Info), and "Gateway Load Balancer" (Info). Each card contains a diagram and a brief description. The "Application Load Balancer" card shows a flow from a client to an ALB, which then routes traffic to three Lambda functions. The "Network Load Balancer" card shows a flow from a client to an NLB, which then routes traffic to three ALBs, each connected to different protocols (TCP, UDP, TLS) and backends. The "Gateway Load Balancer" card shows a flow from a client to a GWLB, which then routes traffic to a third-party virtual appliance. Each card has a "Create" button at the bottom. The page also includes a "Learn more" link and copyright information at the bottom.

Give name to the loadbalancer.

Select Internet facing option

**Create Application Load Balancer** Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

▶ How Application Load Balancers work

**Basic configuration**

**Load balancer name**  
Name must be unique within your AWS account and can't be changed after the load balancer is created.  
 A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

**Scheme** Info  
Scheme can't be changed after the load balancer is created.

**Internet-facing**  
• Serves internet-facing traffic.  
• Has public IP addresses.  
• DNS name resolves to public IPs.  
• Requires a public subnet.

**Internal**  
• Serves internal traffic.  
• Has private IP addresses.  
• DNS name resolves to private IPs.  
• Compatible with the IPv4 and Dualstack IP address types.

**Load balancer IP address type** Info  
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.

**IPv4**  
Includes only IPv4 addresses.

**Dualstack**  
Includes IPv4 and IPv6 addresses.

**Dualstack without public IPv4**  
Includes a public IPv6 address, and private IPv4 and IPv6 addresses. Compatible with **internet-facing** load balancers only.

**Network mapping** Info

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

In Network Mapping section , Select Availability Zones atleast 2.

(choose AZ same as the AZ we selected while creating autoscalling group.)

**IP pools** Info  
You can optionally choose to configure an IPAM pool as the preferred source for your load balancer's IP addresses. Create or view Pools in the Amazon VPC IP Address Manager console [View](#).  
 Use IPAM pool for public IPv4 addresses  
The IPAM pool you choose will be the preferred source of public IPv4 addresses. If the pool is depleted IPv4 addresses will be assigned by AWS.

**Availability Zones and subnets** Info  
Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

**us-east-1a (use1-az1)**  
Subnet  
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.  
 subnet-0e01abced480799  
IPv4 subnet CIDR: 172.31.0.0/20

**us-east-1b (use1-az2)**

**us-east-1c (use1-az4)**  
Subnet  
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.  
 subnet-03b9c870019d28e78  
IPv4 subnet CIDR: 172.31.16.0/20

**us-east-1d (use1-az6)**

**us-east-1e (use1-az3)**

**us-east-1f (use1-az5)**

**Security groups** Info  
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups  
 Select up to 5 security groups

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

In Security Group section , select a existing security group.

(Select same security group we selected while creating autoscalling group)

**Security groups** [Info](#)  
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

**Listeners and routing** [Info](#)  
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

**Default action** [Info](#)  
The default action is used if no other rules apply. Choose the default action for traffic on this listener.

**Routing action**

- Forward to target groups
- Redirect to URL
- Return fixed response

**Forward to target group** [Info](#)  
Choose a target group and specify routing weight or [create target group](#).

Target group	Protocol	Weight	Percent
Home-TG Target type: Instance, IPv4   Target stickiness: Off	HTTP	1	100%

[+ Add target group](#)  
You can add up to 4 more target groups.

In Listners and Routing section

select forward to target group

Choose a default target group. (Here default target group is HomeTG)

no need to change port n protocol.

Click on Create Load Balance

**Summary**  
Review and confirm your configurations. [Estimate cost](#)

**Basic configuration** [Edit](#)  
Name: ALB  
Scheme: Internet-facing  
IP address type: IPv4

**Network mapping** [Edit](#)  
VPC: [vpc-06e769ac5be1c7149](#)  
Public IPv4 IPAM pool: -  
Availability Zones and subnets:

- us-east-1a [subnet-0e01aebcedc480799](#)
- us-east-1c [subnet-03b9c870019d28e78](#)

**Service integrations** [Edit](#)  
Amazon CloudFront + AWS Web Application Firewall (WAF): -  
AWS WAF: -  
AWS Global Accelerator: -

**Attributes**  
Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

**Creation workflow and status**

**Server-side tasks and status**  
After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

[Cancel](#) [Create load balancer](#)

Successfully created Application loadbalancer.

If you scroll down , you will see the HomeTG as default target group in Listeners and Rules section.

To add other Target groups..

select HTTP80 i.e., HomeTG

Click on Manage Rules

Click on Add Rule

The screenshot shows the AWS CloudFront console with the 'Origin' tab selected. The 'Origin Path' dropdown is open, displaying the path '/mobile/\*'. Other tabs visible include 'Behaviors', 'Compress', 'Cache Policy', 'Headers', 'Custom Error Responses', and 'TLS Settings'.

Give name to rule.

In condition section , click on Add condition

The screenshot shows the AWS CloudFront console with the 'Origin' tab selected. The 'Origin Path' dropdown is open, displaying the path '/mobile/\*'. Other tabs visible include 'Behaviors', 'Compress', 'Cache Policy', 'Headers', 'Custom Error Responses', and 'TLS Settings'.

Give path for mobile application..

In Actions section

select forward to target group

select the target group i.e., MobileTG

Go to next step..

In Listners rule , give priority 1 for mobileTG

click on next

**Set rule priority** Info

Each rule has a priority. The default rule is evaluated last. You can change the priority of a non-default rule at any time. You can't change the priority of the default rule.

**Listener rules (2)** Info

Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from the lowest value to the highest value.

Priority	Name tag	Conditions (If)	Actions (Then)	ARN	Tags
1	Mobile-rule	Path = /mobile/*	<ul style="list-style-type: none"> <li>Forward to target group Mobile-TG [1] (100%) Target group stickiness: Off</li> <li>Forward to target group Home-TG [1] (100%) Target group stickiness: Off</li> </ul>	Pending	1 tag
Last (default)	Default	If no other rule applies		ARN	0 tags

**Cancel** **Previous** **Next**

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Click on Add rule.

**Review and create**

**Listener details: HTTP:80**

**Rule details: Mobile-rule**

Priority 1	Conditions If request matches all: Path = /mobile/*	Actions <ul style="list-style-type: none"> <li>Forward to target group Mobile-TG [1] (100%) Target group stickiness: Off</li> </ul>
---------------	---	--

**Rule ARN**  
Pending

**Rule tags (1)**

Tags can help you manage, identify, organize, search for and filter resources.

Key	Value
Name	Mobile-rule

**Edit**

**Server-side tasks and status**

After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

**Cancel** **Previous** **Add rule**

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Successfully connected Mobile Target Group to Loadbalancer

The screenshot shows the AWS CloudFront console with the following details:

- EC2 > Load balancers > ALB > HTTP:80 listener**
- Details:** Successfully created rule "Mobile-rule" on listener HTTP:80.
- Protocol:Port:** HTTP:80
- Load balancer:** ALB
- Default actions:**
  - Forward to target group Home-TG 1 (100%)
  - Target group stickiness: Off
- Listener ARN:** arn:aws:elasticloadbalancing:us-east-1:658394313504:listener/app/ALB/33fb609aff9940ff/b6e86ecaceb9b680
- Rules:** Listener rules (2)

Priority	Name tag	Conditions (If)	Actions (Then)	ARN	Tags	Actions
1	Mobile-rule	Path = /mobile/*	<ul style="list-style-type: none"> <li>Forward to target group Home-TG 1 (100%)</li> <li>Target group stickiness: Off</li> </ul>		1.tag	<i>edit</i> <i>trash</i>
Last (default)	Default	If no other rule applies	<ul style="list-style-type: none"> <li>Forward to target group Home-TG 1 (100%)</li> <li>Target group stickiness: Off</li> </ul>		0 tags	<i>edit</i> <i>trash</i>

Similarly connect Laptop target group to load balancer with priority 2.

The screenshot shows the AWS CloudFront console with the following details:

- EC2 > Load balancers > ALB > HTTP:80 listener > Add rule**
- Step 1:** Add rule (selected)
- Step 2:** Set rule priority
- Step 3:** Review and create
- Add rule Info:** Requests that match this rule are evaluated against its conditions. If a request matches all of the rule's conditions, then the request is routed according to the rule's actions.
- Listener details:** Listener: HTTP:80
- Name and tags:** Name: Laptop-rule
- Conditions (1 value):** Path = /laptop/\*
  - Path condition value: /laptop/\*
  - Valid characters are a-z, A-Z, 0-9 and special characters. Path must be 1-128 characters.
  - + Add OR condition value
  - Add condition

aws [Alt+S] United States (N. Virginia) Shubham

EC2 > Load balancers > ALB > HTTP:80 listener > Add rule

Valid characters are a-z, A-Z, 0-9 and special characters. Path must be 1-128 characters.

+ Add OR condition value

Add condition ▾ You can add up to 4 more condition values for this rule.

**Actions** Info Requests matching all rule conditions route according to the rule actions.

**Routing action**

Forward to target groups  Redirect to URL  Return fixed response

**Forward to target group** Info Choose a target group and specify routing weight or [create target group](#).

**Target group**

Laptop-TG	HTTP	Weight	Percent
Target type: Instance, IPv4   Target stickiness: Off		1 0-999	100%

+ Add target group You can add up to 4 more target groups.

**Target group stickiness** Info Enables the load balancer to bind a user's session to a specific target group. To use stickiness the client must support cookies. If you want to bind a user's session to a specific target, turn on the Target Group attribute Stickiness.

Turn on target group stickiness

**Cancel** **Next**

The screenshot shows the AWS CloudWatch Metrics Insights interface. At the top, there's a search bar and navigation links for 'Search' and '[Alt+S]'. The main header reads 'Metrics Insights' with a 'Create new' button. Below the header, a breadcrumb trail shows the path: 'Metrics Insights' > 'Metrics' > 'CloudWatch Metrics' > 'Metrics Insights'. On the left, a sidebar lists 'Step 1' (Add rule), 'Step 2' (Set rule priority), and 'Step 3' (Review and create). The 'Set rule priority' step is currently selected. The main content area has a title 'Set rule priority' with an 'Info' link. A note states: 'Each rule has a priority. The default rule is evaluated last. You can change the priority of a non-default rule at any time. You can't change the priority of the default rule.' Below this is a section titled 'Listener details: HTTP:80' with a 'Listener rules (3)' link and an 'Info' link. A table lists three rules: 'Laptop-rule' (Priority 2, Path: /laptop/\*), 'Mobile-rule' (Priority 1, Path: /mobile/\*), and 'Last (default)' (Path: If no other rule applies). Each row shows actions like 'Forward to target group' and target group details. Buttons for 'Rule limits', 'Reset priorities', and 'Add gap between priorities' are also present.

CloudShell   Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Account ID: 6583-9431-3304 ▾ Shubham

aws Search [Alt+S]

EC2 > Load balancers > ALB > HTTP:80 listener > Add rule

Step 1  
 Add rule  
 Step 2  
 Set rule priority  
 Step 3  
 Review and create

## Review and create

▶ Listener details: HTTP:80

▼ Rule details: Laptop-rule

Priority 2	Conditions If request matches all: Path = <code>/laptop/*</code>	Actions <ul style="list-style-type: none"> <li>Forward to target group Laptop-TG  1 (100%) Target group stickiness: Off</li> </ul>
Rule ARN <i>Pending</i>		

**Rule tags (1)** 

Tags can help you manage, identify, organize, search for and filter resources.

Key	Value
Name	Laptop-rule

▶ Server-side tasks and status

After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

Now if you check in instance dashboard you will see the Active instances got created automatically.

The screenshot shows the AWS EC2 Instances dashboard. On the left, there's a navigation sidebar with sections like Dashboard, EC2 Global View, Events, Instances (with sub-options like Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs). The main content area is titled 'Instances (8) Info' and contains a table with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, Public IPv4 DNS, Public IPv4, and Elastic IP. All instances are listed as 'Running'. Below the table, there's a section titled 'Select an instance' with a dropdown menu.

To check if it is working properly

go to Load balancer.

click on the load balancer we created

scroll down and copy DNS name

search it through browser

The screenshot shows the AWS Load Balancers dashboard. The left sidebar includes sections for Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), and Auto Scaling (Auto Scaling Groups, Settings). The main area displays a table for 'Load balancers (1/1)' with one entry named 'ALB'. It shows details like State (Active), Type (application), Scheme (Internet-facing), IP address type (IPv4), VPC ID (vpc-06e769ac5be1c7149), Availability Zones (2), Security groups (sg-0e1ca5a74f9e79c5f), and DNS name (ALB-1898). Below this, there's a detailed view for the 'ALB' load balancer, showing tabs for Details, Listeners and rules, Network mapping, Resource map, Security, Monitoring, Integrations, Attributes, Capacity, and Tags. The 'Details' tab is selected, displaying information such as Load balancer type (Application), Status (Active), Scheme (Internet-facing), Hosted zone (Z35SXDOTRQ7X7K), VPC (vpc-06e769ac5be1c7149), Availability Zones (us-east-1a, us-east-1c), Date created (September 27, 2025, 19:12 (UTC+0:30)), and DNS name (ALB-1898267340.us-east-1.elb.amazonaws.com (A Record)).

## Step6: Results

The ALB DNS successfully routed traffic to all three target groups.

Scaling actions were verified:

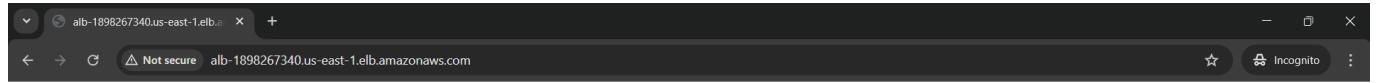
Home ASG maintained fixed instances.

Mobile ASG scaled according to schedule.

Laptop ASG scaled dynamically on CPU utilization.

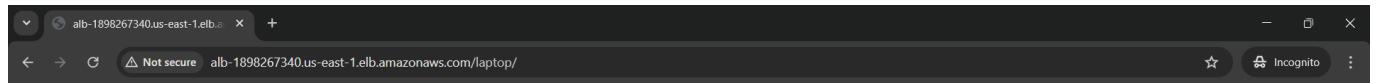
Application remained available across multiple Availability Zones.

Home:



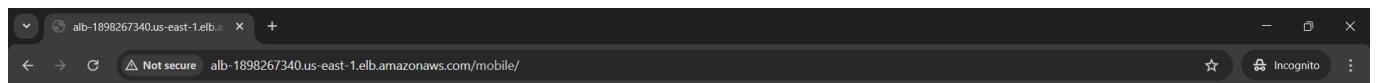
**This is my home page ip-172-31-31-118.ec2.internal**

Laptop:



**This is my laptop page ip-172-31-9-140.ec2.internal**

Mobile:



**This is my mobile page ip-172-31-25-40.ec2.internal**

## Technologies Used

Amazon EC2 – Virtual servers to host the application.

Application Load Balancer (ALB) – For distributing incoming traffic across target groups.

Launch Templates

Auto Scaling Groups (ASG) – To implement different scaling strategies (Static, Dynamic, Scheduled).

Target Groups – For routing traffic to the right set of instances.

Amazon Linux – Operating system used for EC2 instances.

Security Groups – For controlling inbound and outbound traffic.

## Conclusion

This project shows how to build a website on AWS that is reliable, can handle lots of visitors, and fixes itself automatically. I got hands-on experience by setting up three different ways to automatically add or remove servers:

Keeping a fixed number running.

Adding more at scheduled busy times.

Letting AWS add or remove them based on current traffic.

All traffic entered through one web address, and the system automatically directed visitors to available servers. It successfully adjusted the number of servers and handled all the web requests.

This project gave me a strong, practical understanding of how to manage traffic and automate server scaling on AWS.